

Mid-Semester Check In 2

2025-07-19

Load Libraries

```
#Always good to get all the packages up and running before doing anything else
library(data.table)
library(ggplot2)
library(tidyverse)
```

```
## — Attaching core tidyverse packages ————— tidyverse 2.0.0 —
## ✓ dplyr      1.1.4      ✓ readr      2.1.5
## ✓ forcats    1.0.0      ✓ stringr    1.5.1
## ✓ lubridate  1.9.4      ✓ tibble     3.2.1
## ✓ purrr      1.0.2      ✓ tidyr      1.3.1
## — Conflicts ————— tidyverse_conflicts() —
## ✗ dplyr::between() masks data.table::between()
## ✗ dplyr::filter()  masks stats::filter()
## ✗ dplyr::first()   masks data.table::first()
## ✗ lubridate::hour() masks data.table::hour()
## ✗ lubridate::isoweek() masks data.table::isoweek()
## ✗ dplyr::lag()      masks stats::lag()
## ✗ dplyr::last()     masks data.table::last()
## ✗ lubridate::mday() masks data.table::mday()
## ✗ lubridate::minute() masks data.table::minute()
## ✗ lubridate::month() masks data.table::month()
## ✗ lubridate::quarter() masks data.table::quarter()
## ✗ lubridate::second() masks data.table::second()
## ✗ purrr::transpose() masks data.table::transpose()
## ✗ lubridate::wday()  masks data.table::wday()
## ✗ lubridate::week()  masks data.table::week()
## ✗ lubridate::yday()  masks data.table::yday()
## ✗ lubridate::year()  masks data.table::year()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

Load metadata and gene expression data and

created a combined data frame.

```
filename <- "/Users/nupoormarwah/Downloads/QBS103_GSE157103_series_matrix-1.csv"
meta_data <- fread(filename)

filename <- "/Users/nupoormarwah/Downloads/QBS103_GSE157103_genes.csv"

#Makes so that V1 col that contains the gene names are now the row names
gene_expr <- fread(filename) %>%
  column_to_rownames("V1") %>%
#Transposing with make the table be 126 observations by 100 observations
  t() %>%
  data.frame()

#Create combined data frame. Now we have one single data frame with all the data
comb_df <- cbind(meta_data, gene_expr)
```

#Started making plots

#Created a function with four parameters. The first one was df, second genes, etc. The df was for data frame, second was for the actual gene list, continuous_cov was for the numeric covariate Charlson score, and cat_cov was for our categorical covariates sex and disease status.

```
create_gene_plots <- function(df, genes, continuous_cov, cat_cov){
  #Having strings passed into ggplot causes an error. We can use "!!sym()" to refer to a column when its name is stored as a string. !! unquotes the symbol. We used lapply to create a loop that let me create plots for each gene.
```

```
  hist_plots <- lapply(genes, function(gene){
    ggplot(data = df, mapping = aes(x = !!sym(gene))) +
    geom_histogram(fill = "purple", col = "black") +
    #Paste helps combine strings
    labs(x = paste(gene, "Expression"), y = "Count", title = paste("Histogram of", gene, "Gene Expression")) +
    theme_bw() +
    theme(
      plot.title = element_text(face = "bold", hjust = 0.50)
    )
  })
```

#Scatterplots

#"str_to_title" used to capitalize the first letter of each word. "gsub" used to substitute spaces for underscores.

```
cont_cov_label <- str_to_title(gsub("_", " ", continuous_cov))
#Similar to the histogram, used lapply to create a loop that let me create plots for each gene.
```

```
scatter_plots <- lapply(genes, function(gene){
  ggplot(data = df, mapping = aes(x = !!sym(gene), y = !!sym(continuous_cov))) +
  geom_point(color = "blue", size = 2) +
  labs(
    x = paste(gene, "Expression"), y = cont_cov_label,
    title = paste("Scatterplot of", cont_cov_label, "vs", gene, "Expression")
  ) +
  theme_bw() +
  theme(
    plot.title = element_text(face = "bold", hjust = 0.50)
  )
})
```

#Boxplots

#Again used "str_to_title" and "gsub" to clean up the title names for sex and disease status.

```
cat_cov_label1 <- str_to_title(gsub("_", " ", cat_cov[1]))
cat_cov_label2 <- str_to_title(gsub("_", " ", cat_cov[2]))
box_plots <- lapply(genes, function(gene){
  ggplot(data = df, mapping = aes(x = !!sym(cat_cov[1]), y = !!sym(gene), fill = !!sym(cat_cov[1]))) +
  geom_boxplot() +
  #Had to replace the ~ with the vars function for the faceting to work.
  facet_wrap(vars(!!sym(cat_cov[2]))) +
  labs(
```

```

    x = cat_cov_label1, y = paste(gene, "Expression"),
    title = paste("Boxplot of", gene, "Expression by", cat_cov_label1, "and", cat_cov_
label2),
    fill = cat_cov_label1
  ) +
  theme_bw() +
  theme(
    plot.title = element_text(face = "bold", hjust = 0.50)
  )
})

#Created a list of things to return.
return(list(
  "Histograms" = hist_plots, "Scatterplots" = scatter_plots, "Boxplots" = box_plots
))
}

#Doing it this way will update the title, axis, and gene that was plotted. Assigned the
output of our function ot a variable named res1. You should see res1 in our environment.
If you print res1, you'll get all the plots.
res1 <- create_gene_plots(
  df = comb_df, genes = c("A2M", "AASDHPPT", "AAAS"),
  continuous_cov = "charlson_score",
  cat_cov = c("sex", "disease_status")
)

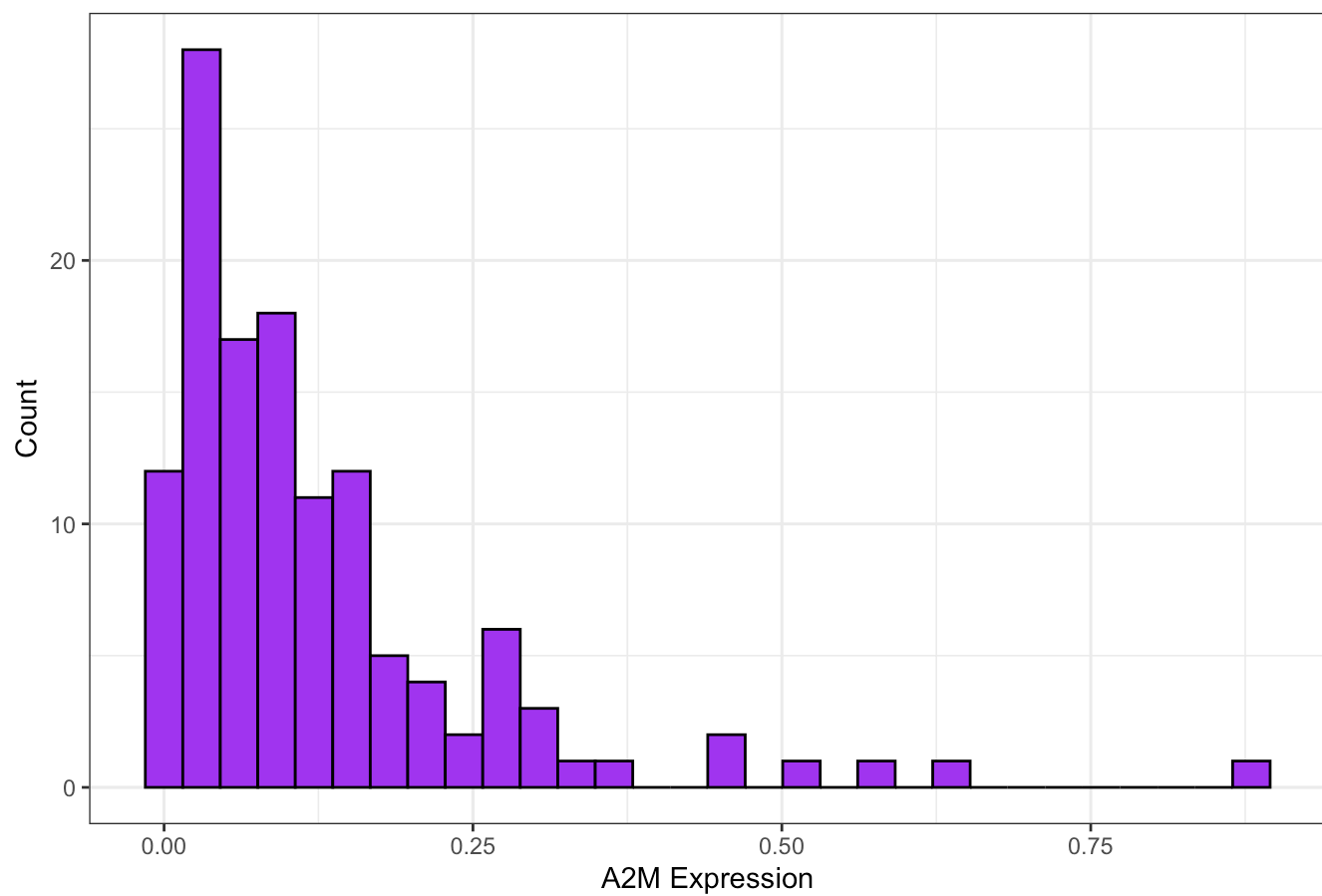
#Printing histograms, scatterplots, and boxplots separately.
res1[["Histograms"]]

```

```
## [[1]]
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

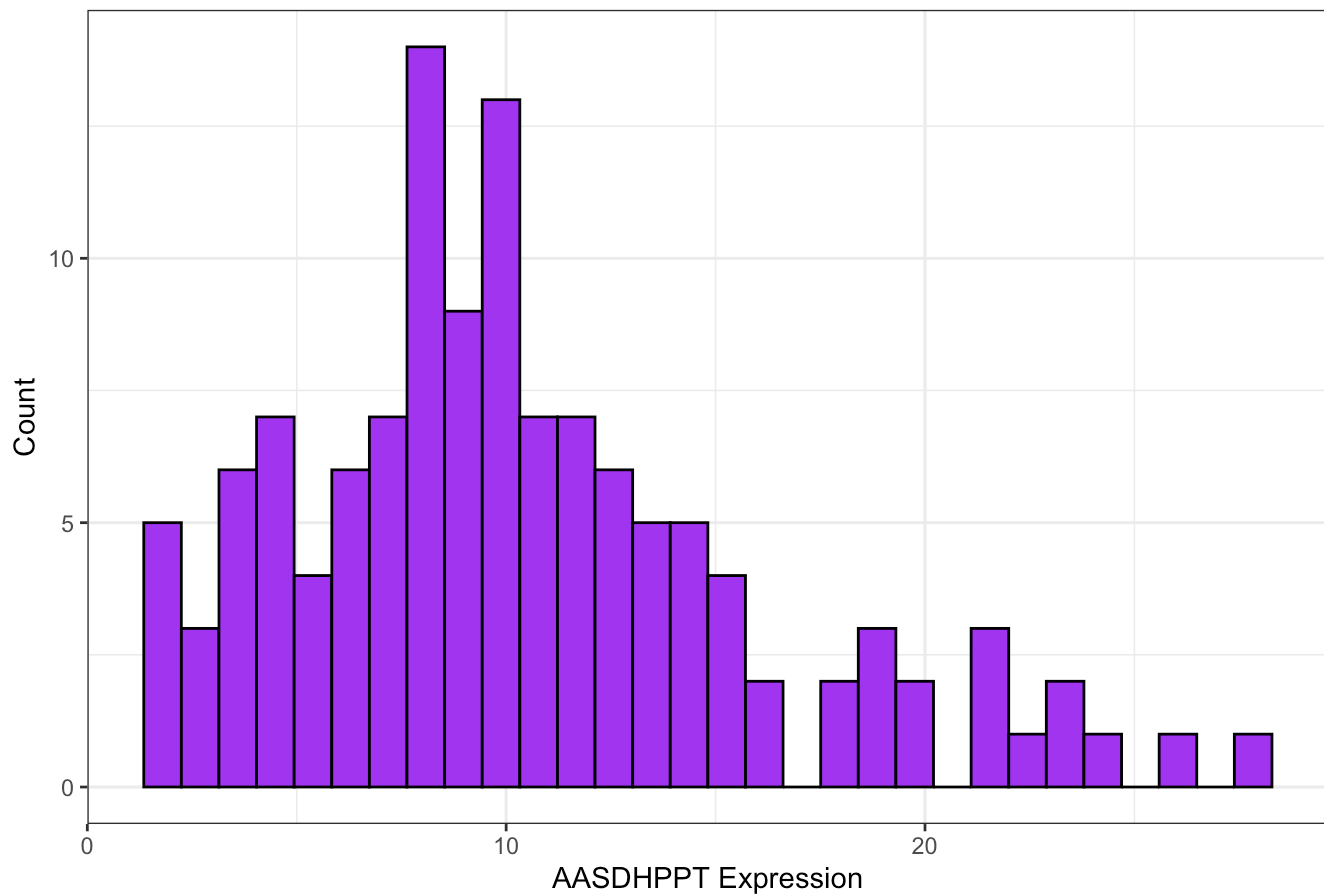
Histogram of A2M Gene Expression



```
##  
## [[2]]
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

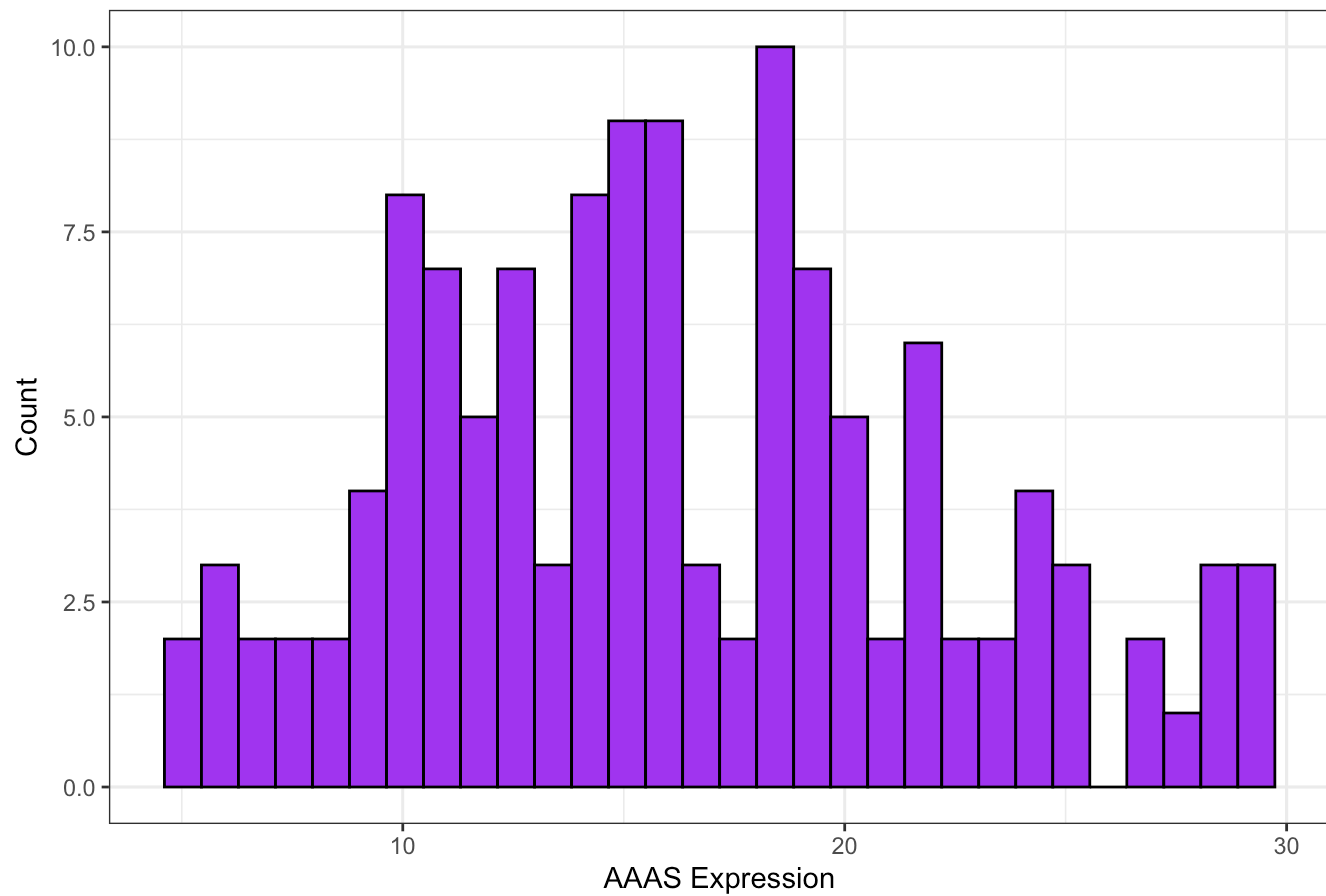
Histogram of AASDHPPT Gene Expression



```
##  
## [[3]]
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

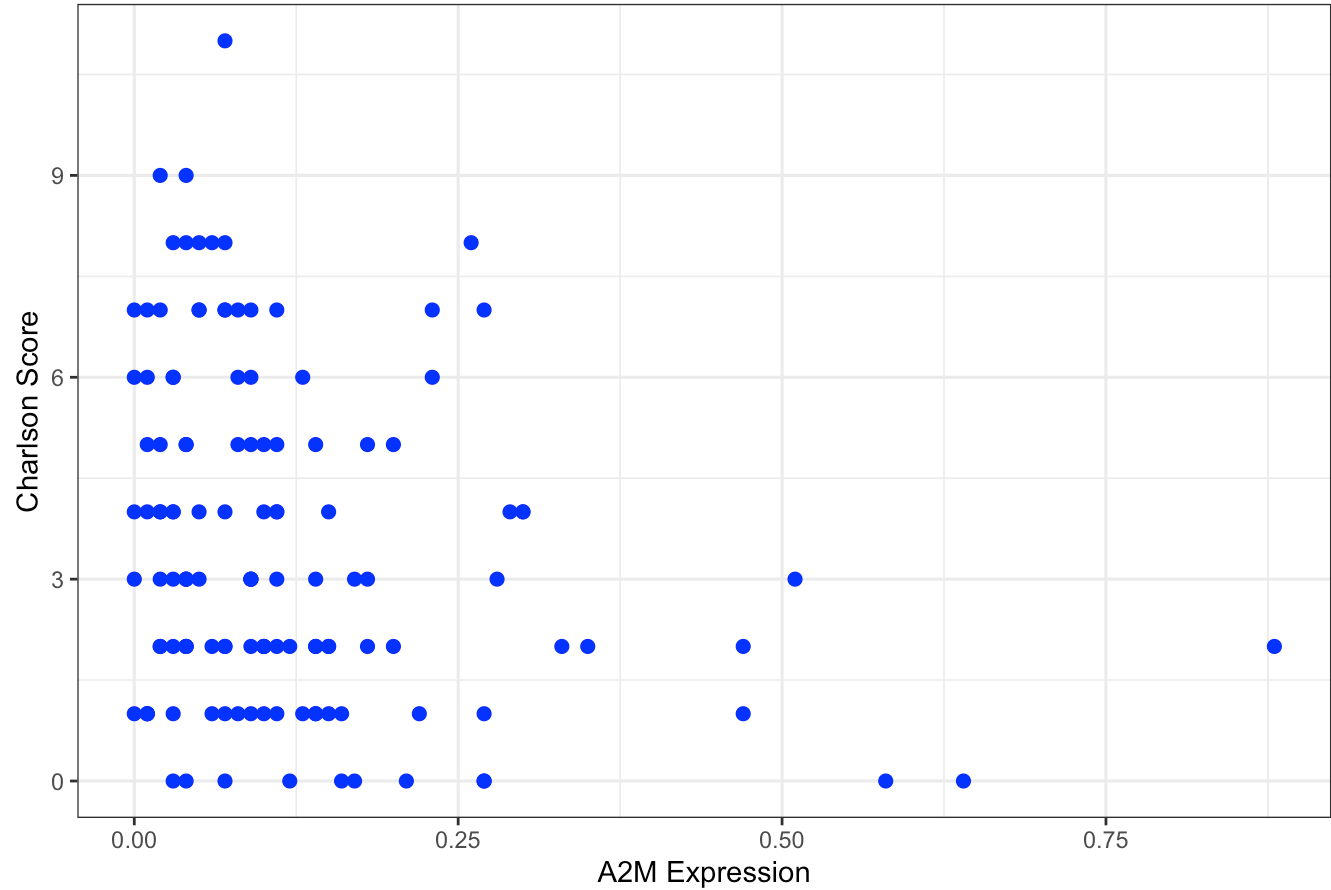
Histogram of AAAS Gene Expression



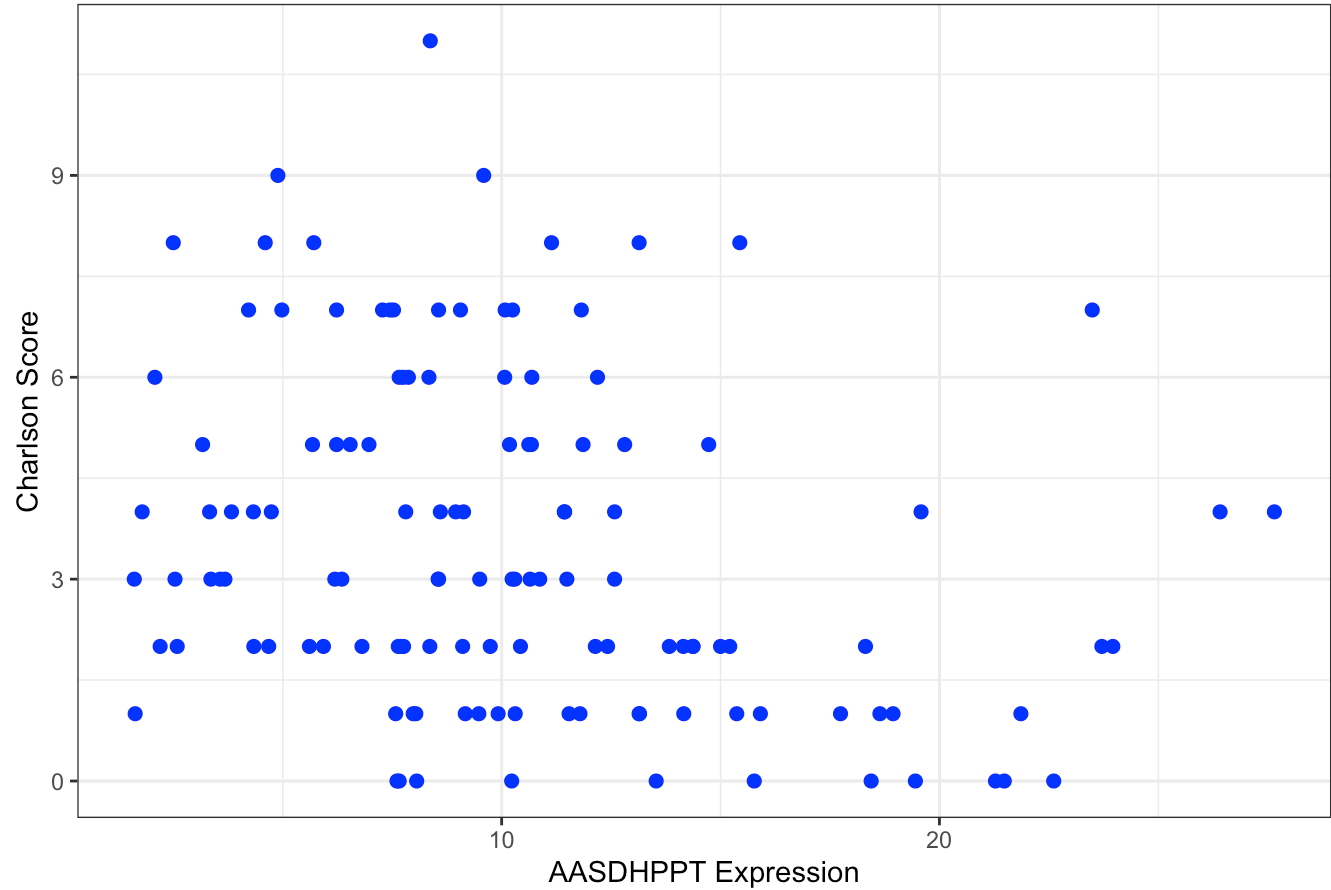
```
res1[["Scatterplots"]]
```

```
## [[1]]
```

Scatterplot of Charlson Score vs A2M Expression

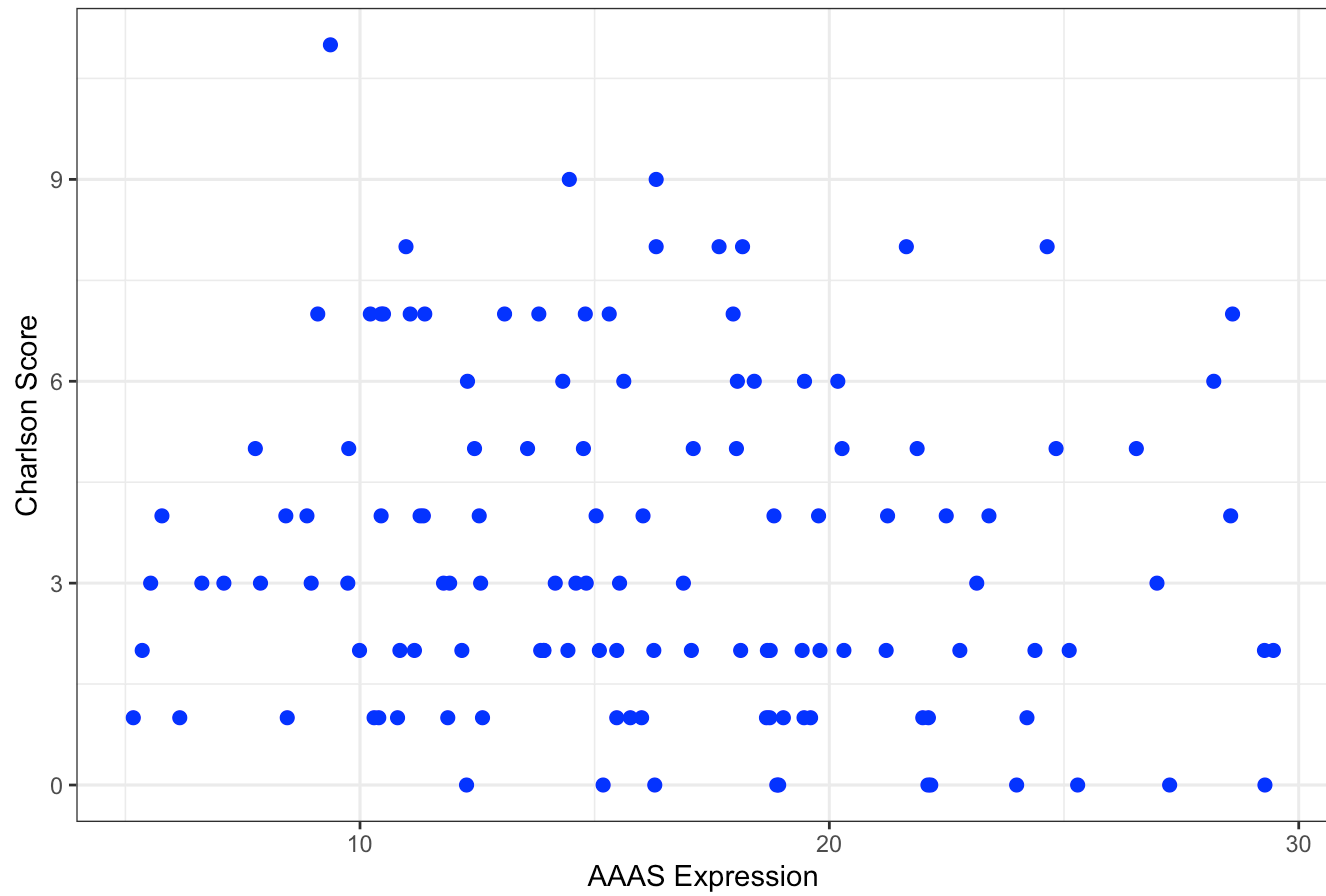


Scatterplot of Charlson Score vs AASDHPPT Expression



```
##  
## [[3]]
```

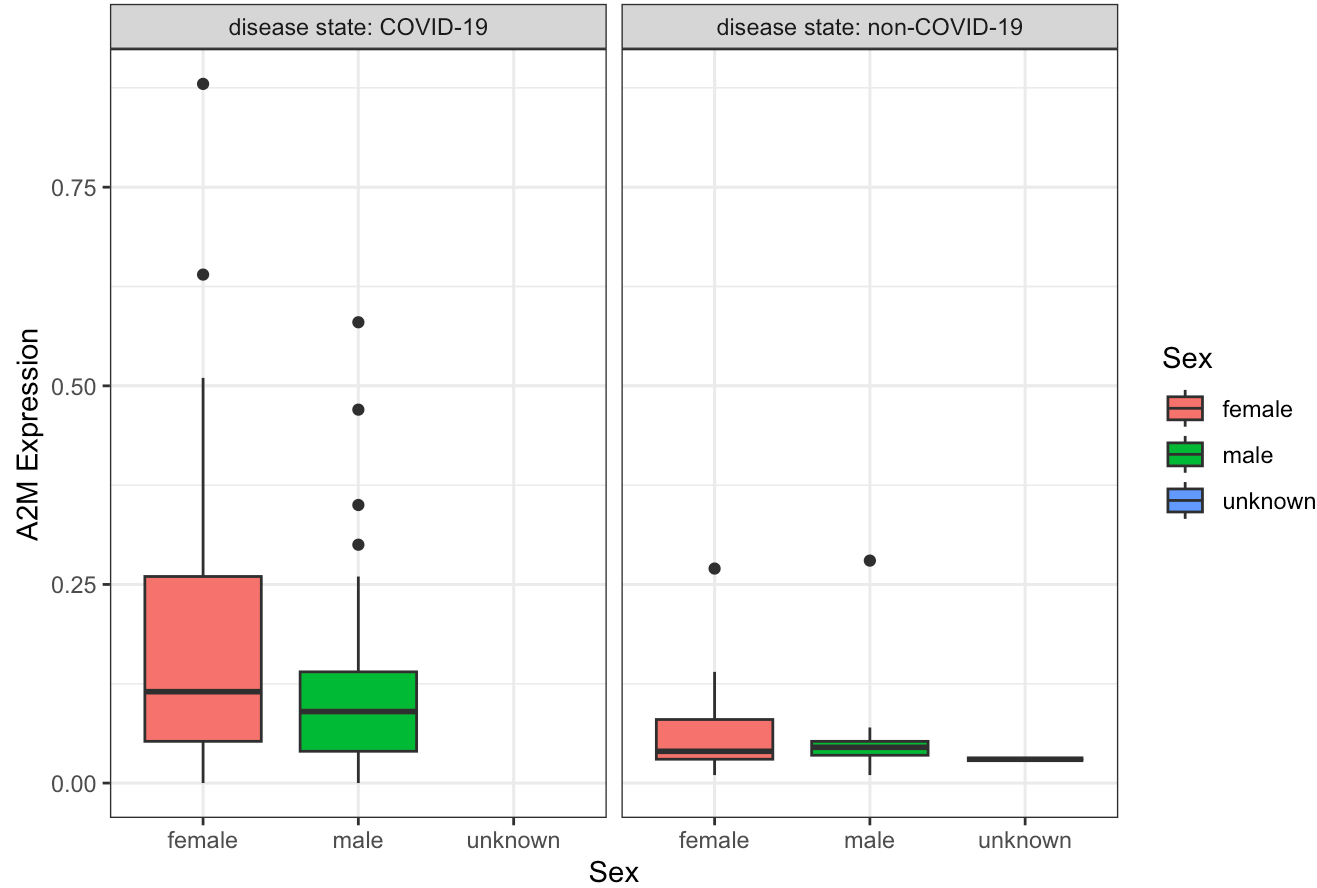
Scatterplot of Charlson Score vs AAAS Expression



```
res1[["Boxplots"]]
```

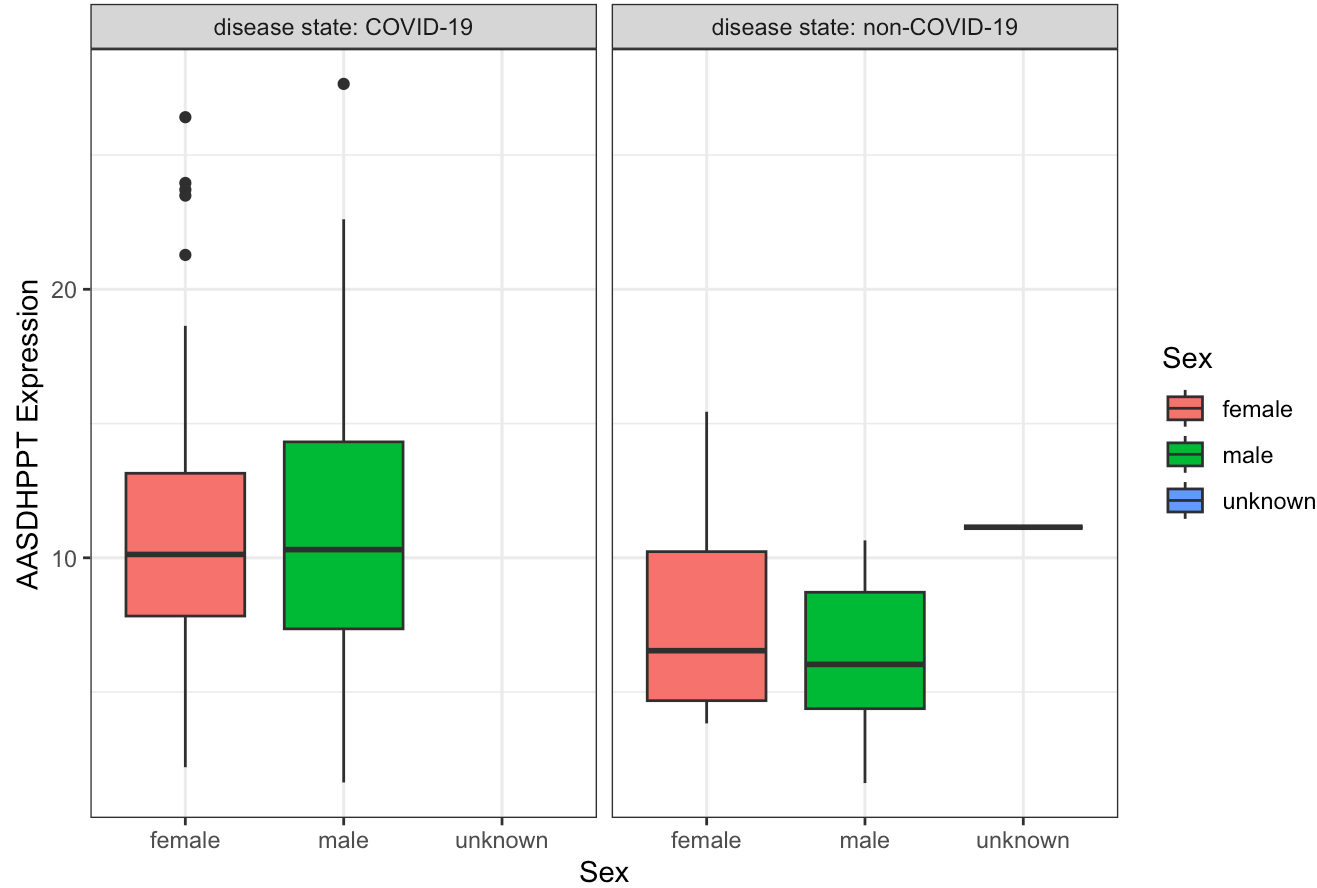
```
## [[1]]
```

Boxplot of A2M Expression by Sex and Disease Status



```
##  
## [[2]]
```

Boxplot of AASDHPPT Expression by Sex and Disease Status



```
##  
## [[3]]
```

Boxplot of AAAS Expression by Sex and Disease Status

